



FIGURE 1

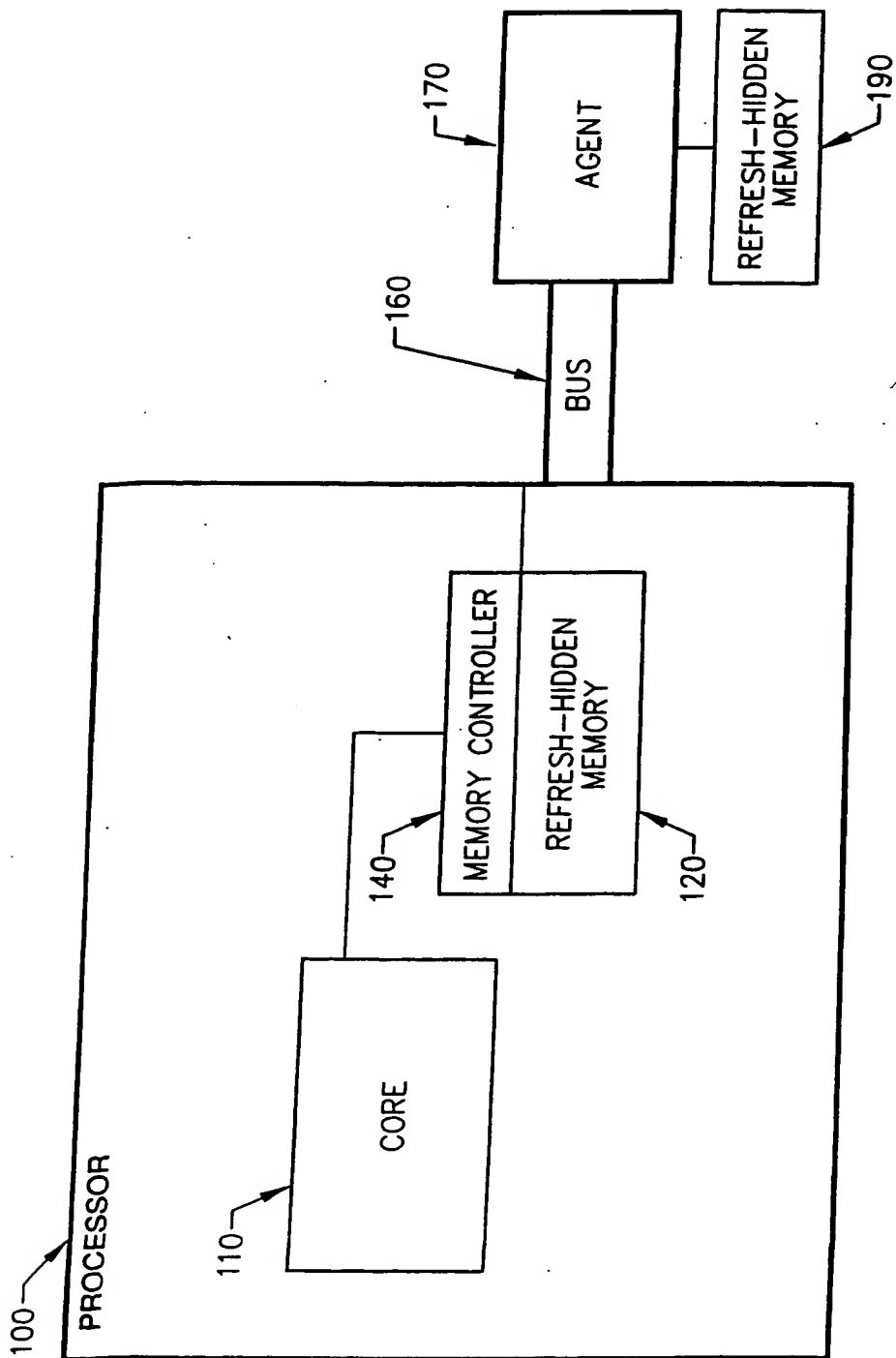


FIGURE 2

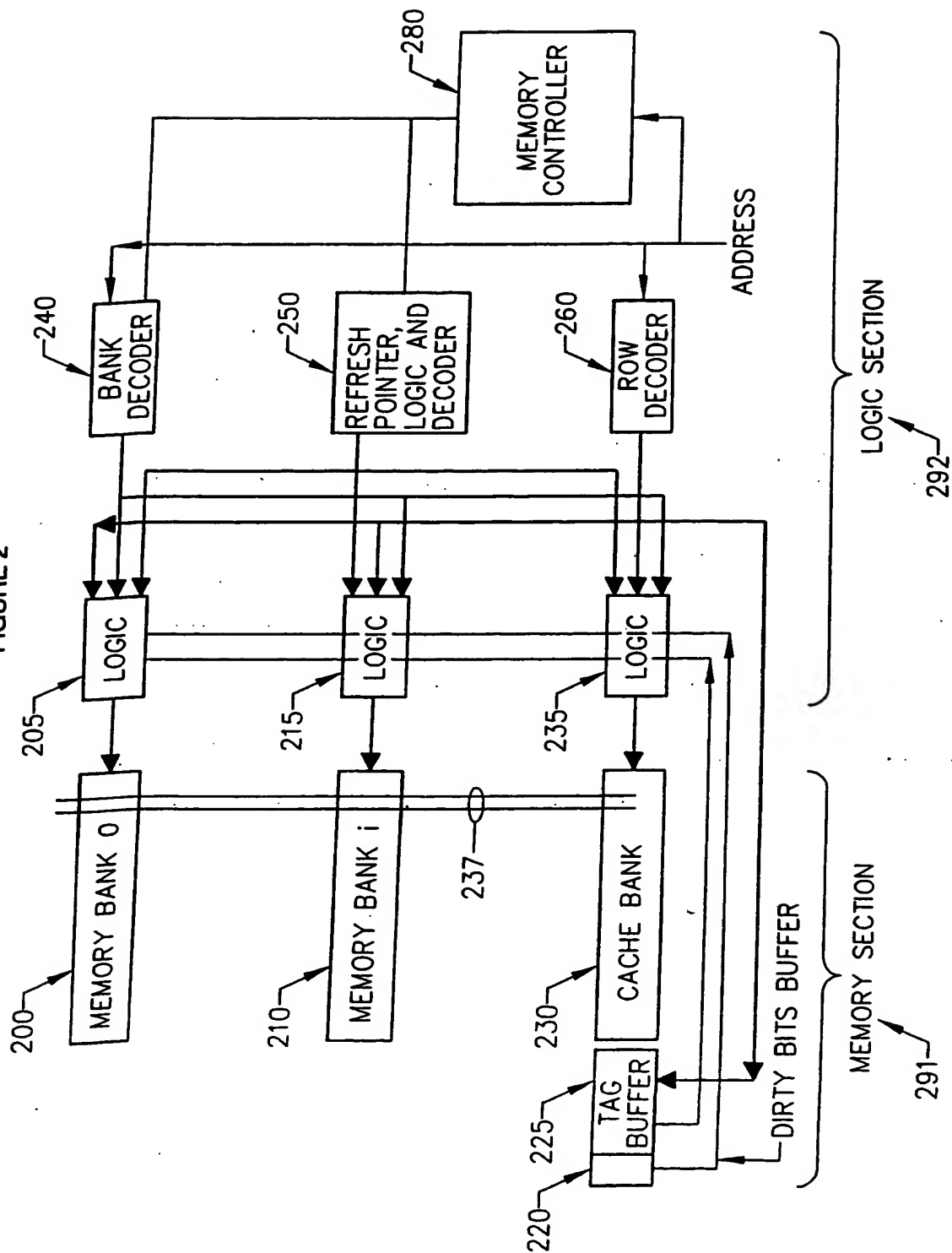
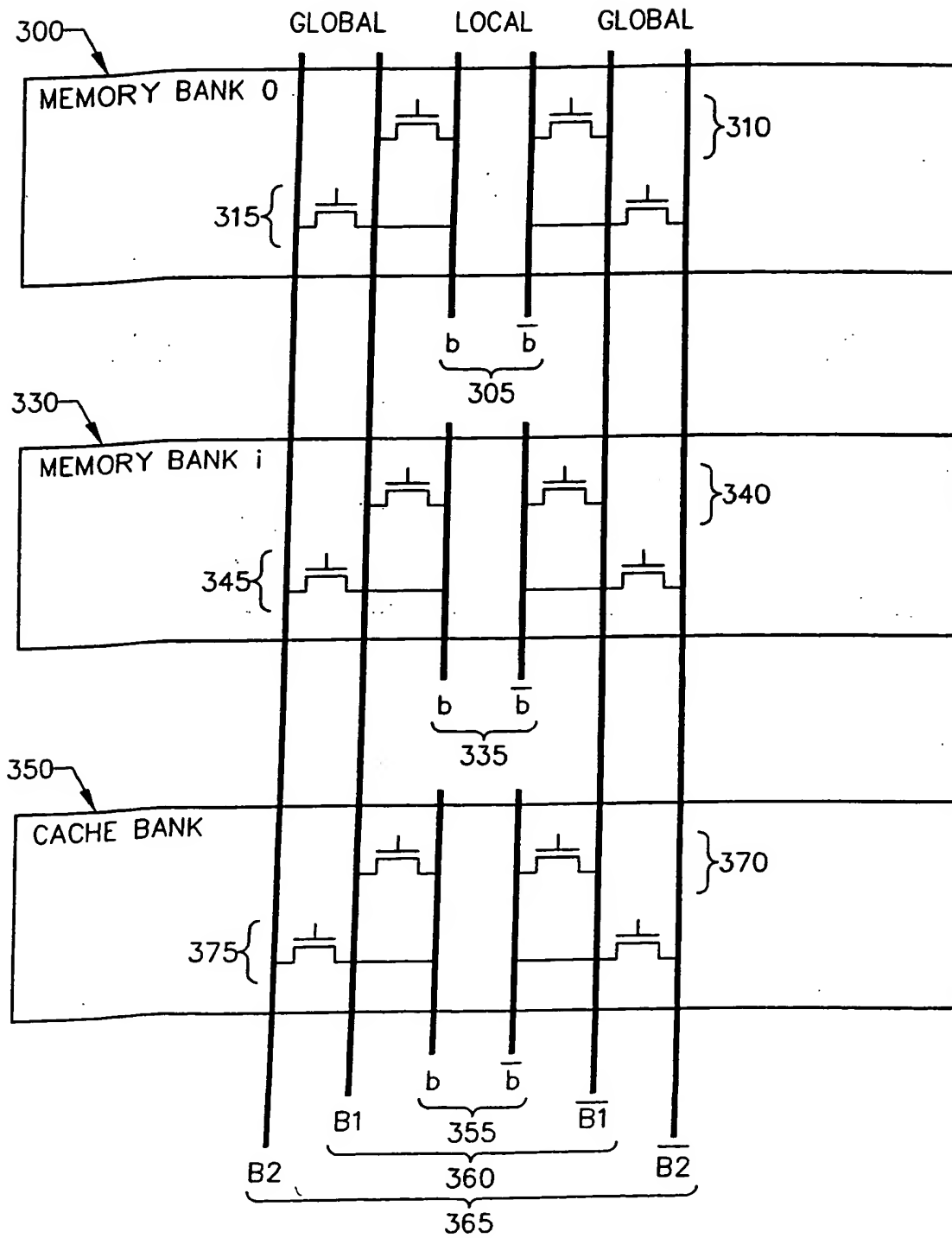


FIGURE 3



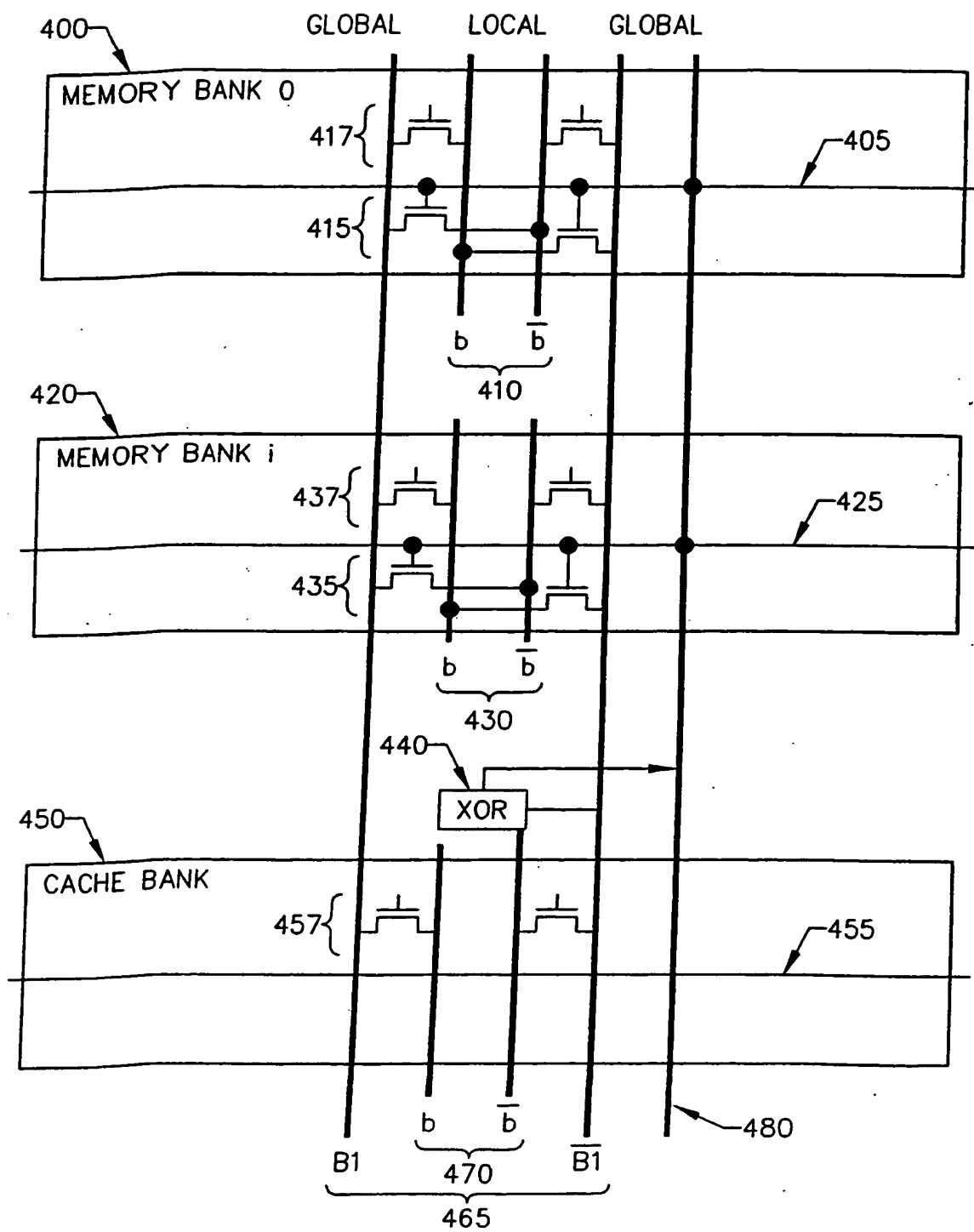


FIGURE 5

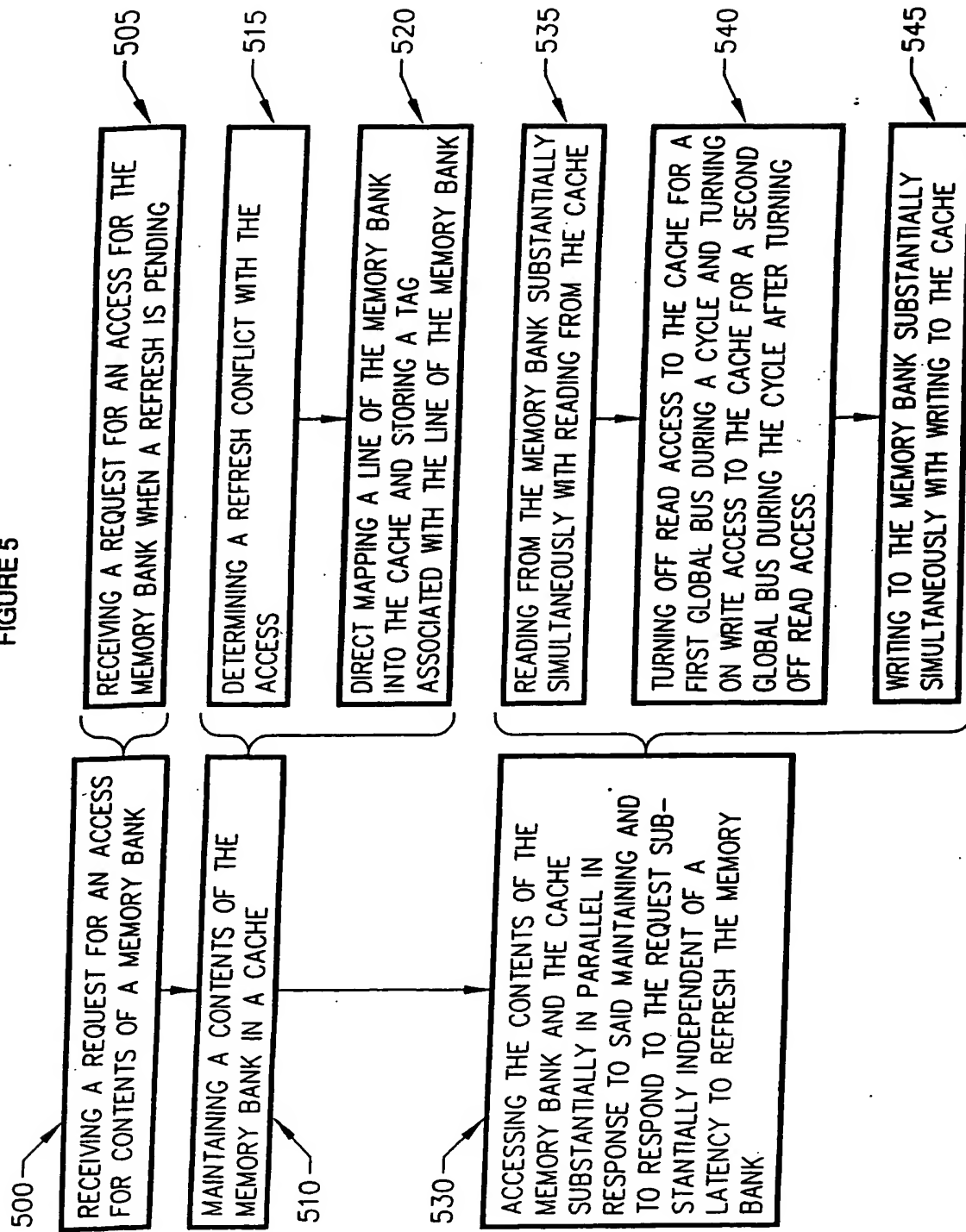
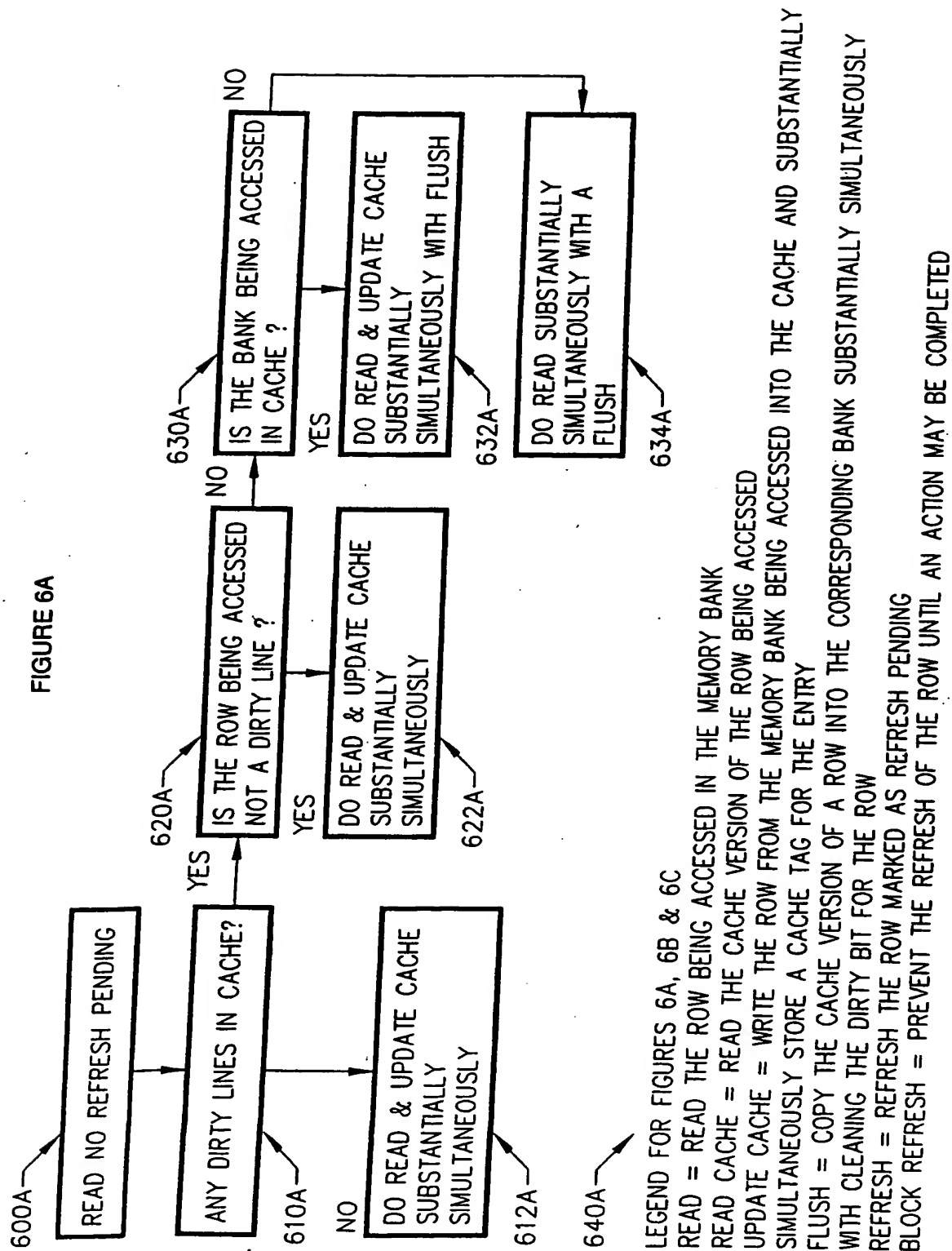
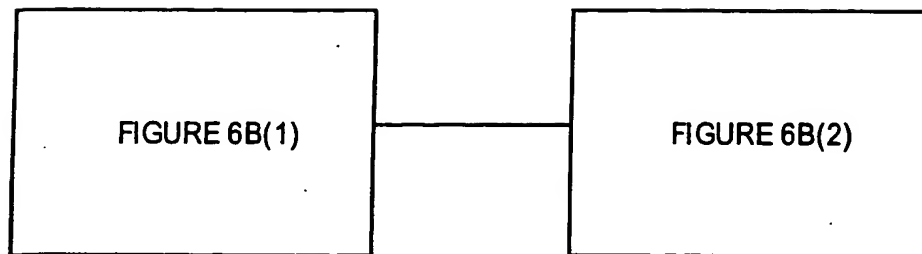


FIGURE 6A





**FIG. 6B**

FIGURE 6B(1)

```
graph TD
    600B[600B] --> 610B{610B: ANY DIRTY LINES IN CACHE?}
    610B -- YES --> 612B{612B: IF BANK MARKED WITH REFRESH PENDING IS NOT THE BANK BEING ACCESSED THEN DO READ SUBSTANTIALLY SIMULTANEOUSLY WITH REFRESH;}
    612B -- YES --> 614B{614B: OTHERWISE BANK MARKED FOR REFRESH IS THE BANK BEING ACCESSED SO IF BANK BEING ACCESSED IS IN CACHE THEN DO READ CACHE SUBSTANTIALLY SIMULTANEOUSLY WITH REFRESH;}
    614B -- YES --> 616B{616B: OTHERWISE IF ROW BEING ACCESSED IS THE ROW MARKED WITH REFRESH PENDING THEN DO READ & UPDATE CACHE;}
    616B -- YES --> 618B{618B: OTHERWISE DO BLOCK REFRESH SUBSTANTIALLY SIMULTANEOUSLY WITH READ & UPDATE CACHE}
    618B --> 610B
    610B -- NO --> 630B{630B: IS THE BANK MARKED FOR REFRESH NOT THE BANK BEING ACCESSED ?}
    630B -- YES --> 632B{632B: DO REFRESH SUBSTANTIALLY SIMULTANEOUSLY WITH THE FOLLOWING OPERATION(S). IS THE BANK STORED IN THE CACHE BANK NOT THE BANK BEING ACCESSED ?}
    632B -- YES --> 634B{634B: IF ROW BEING ACCESSED CORRESPONDS TO A DIRTY LINE THEN DO READ SUBSTANTIALLY SIMULTANEOUSLY WITH FLUSH; OTHERWISE DO READ & UPDATE CACHE}
    634B --> 636B{636B: IF ROW BEING ACCESSED A DIRTY LINE THEN DO READ CACHE SUBSTANTIALLY SIMULTANEOUSLY WITH FLUSH; OTHERWISE DO READ}
    636B --> 610B
```

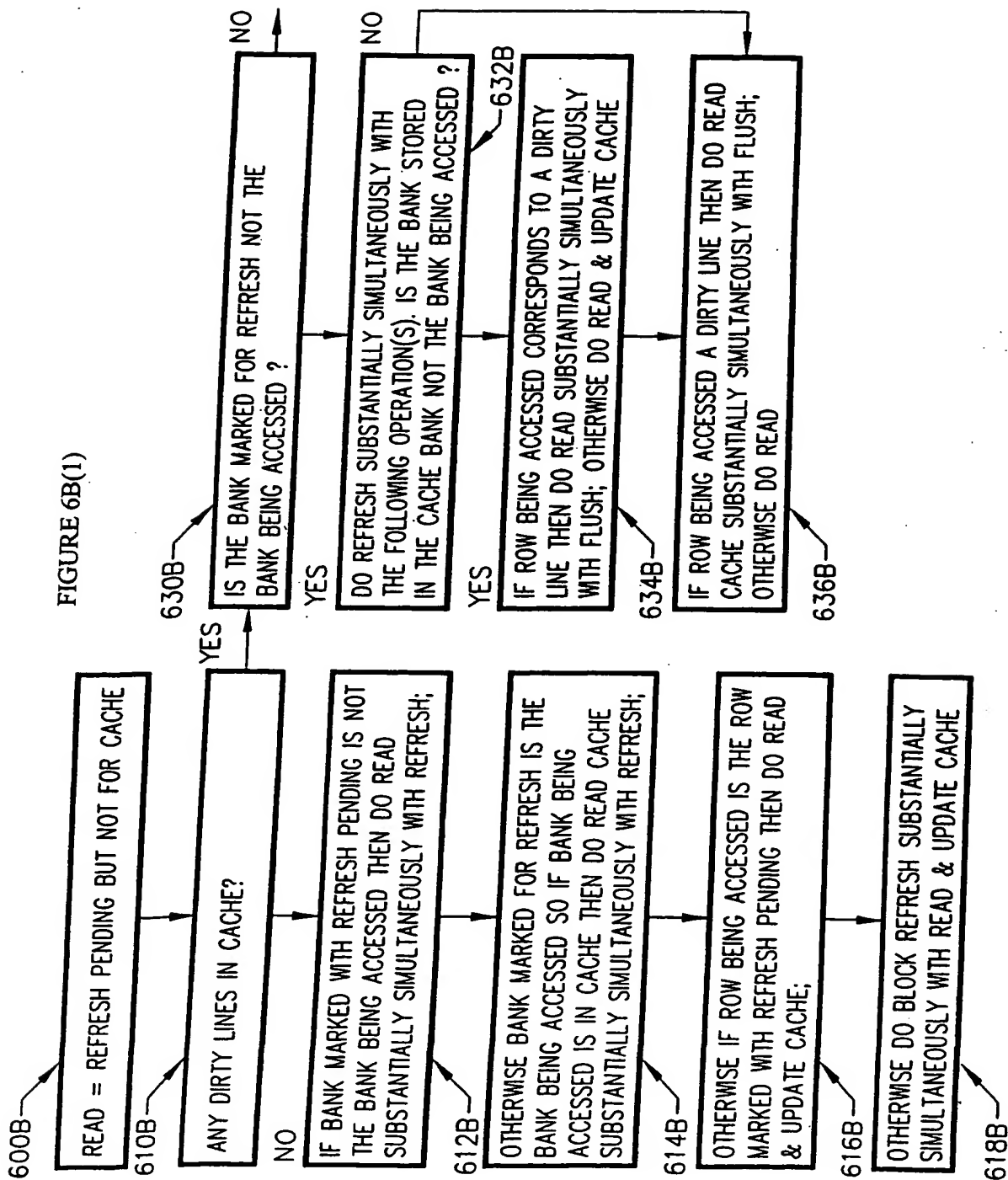




FIGURE 6B(2)

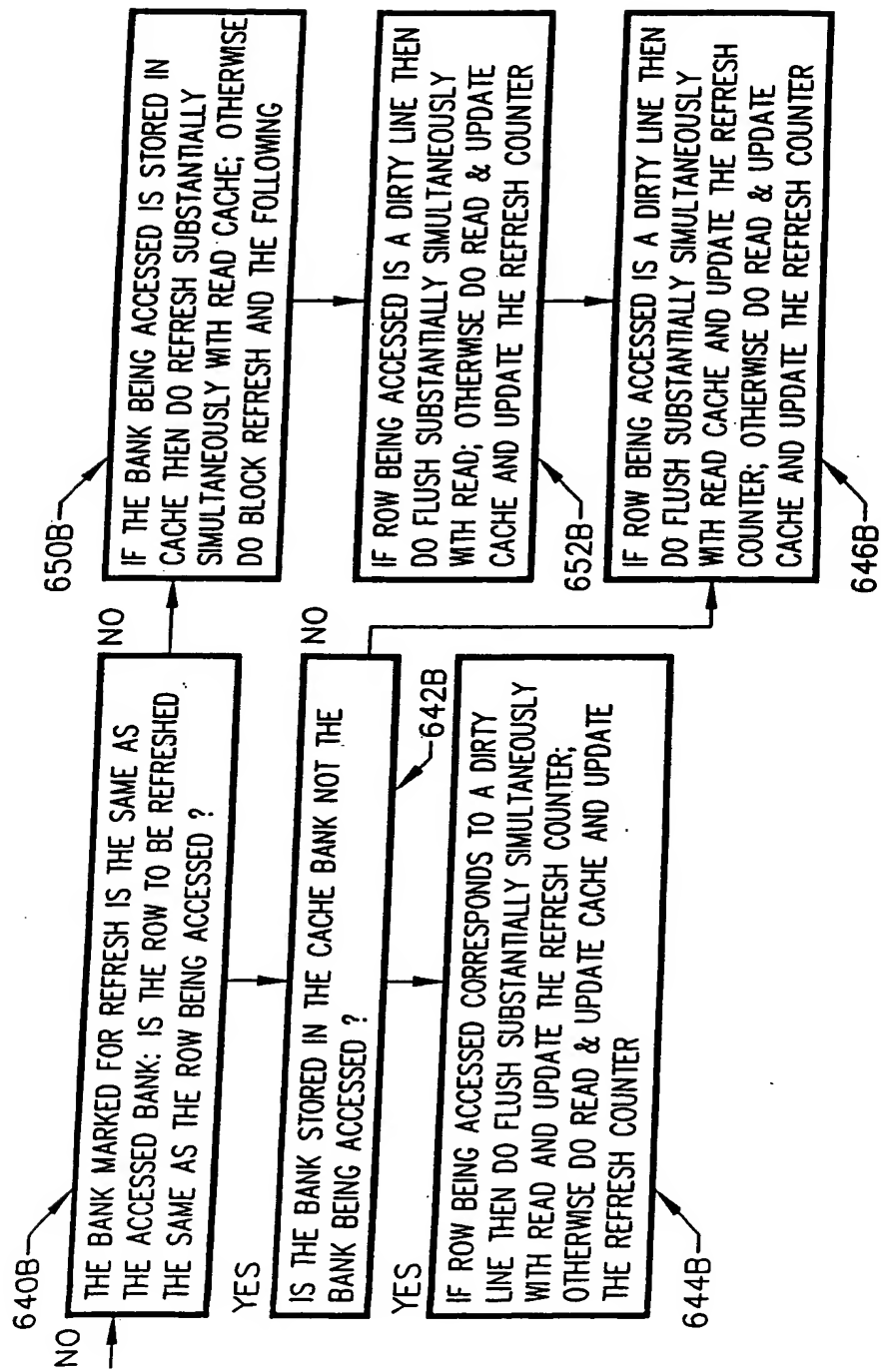
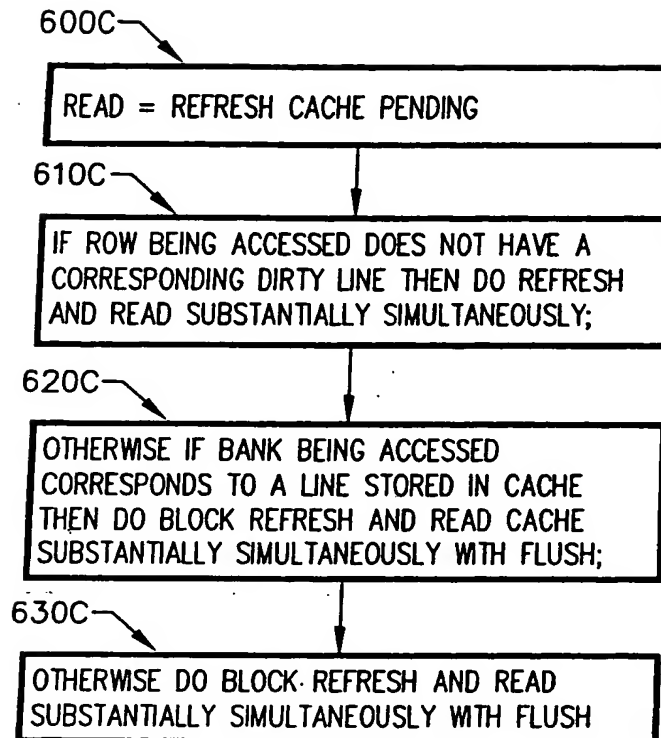
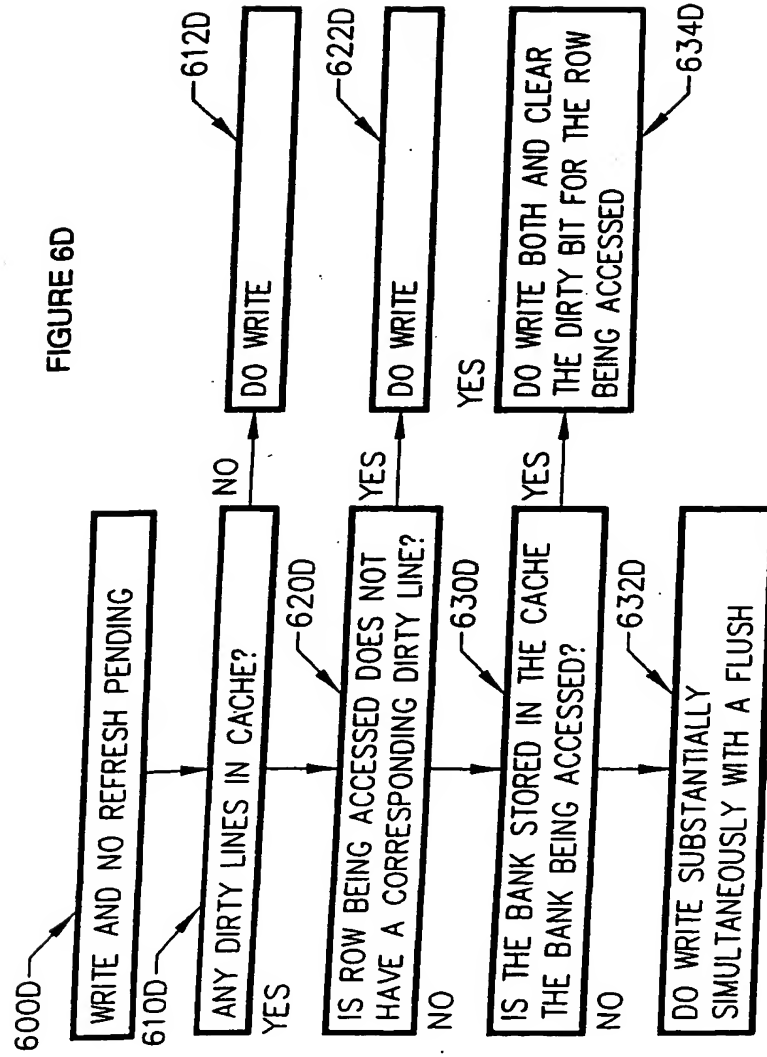


FIGURE 6C





640D

LEGEND FOR FIGURES 6D, 6E & 6F

WRITE = WRITE TO THE ROW BEING ACCESSED IN THE MEMORY BANK

WRITE CACHE = WRITE TO THE CACHE VERSION OF THE ROW BEING ACCESSED

UPDATE CACHE = WRITE THE ROW FROM THE MEMORY BANK BEING ACCESSED INTO THE CACHE AND SUBSTANTIALLY SIMULTANEOUSLY STORE A CACHE TAG FOR THE ENTRY

FLUSH = COPY THE CACHE VERSION OF A ROW INTO THE CORRESPONDING BANK SUBSTANTIALLY SIMULTANEOUSLY WITH CLEANING THE DIRTY BIT FOR THE ROW

REFRESH = REFRESH THE ROW MARKED AS REFRESH PENDING

BLOCK REFRESH = PREVENT THE REFRESH OF THE ROW UNTIL AN ACTION MAY BE COMPLETED

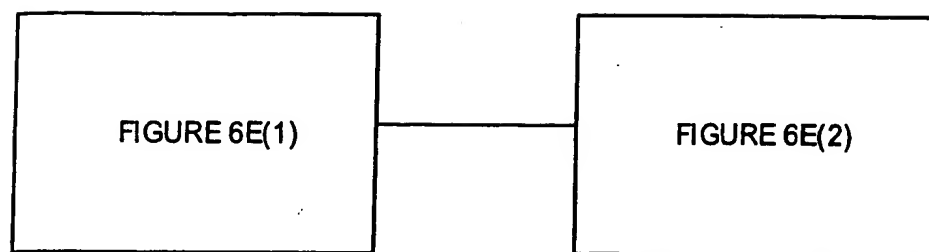


FIG. 6E

FIGURE 6E(1)

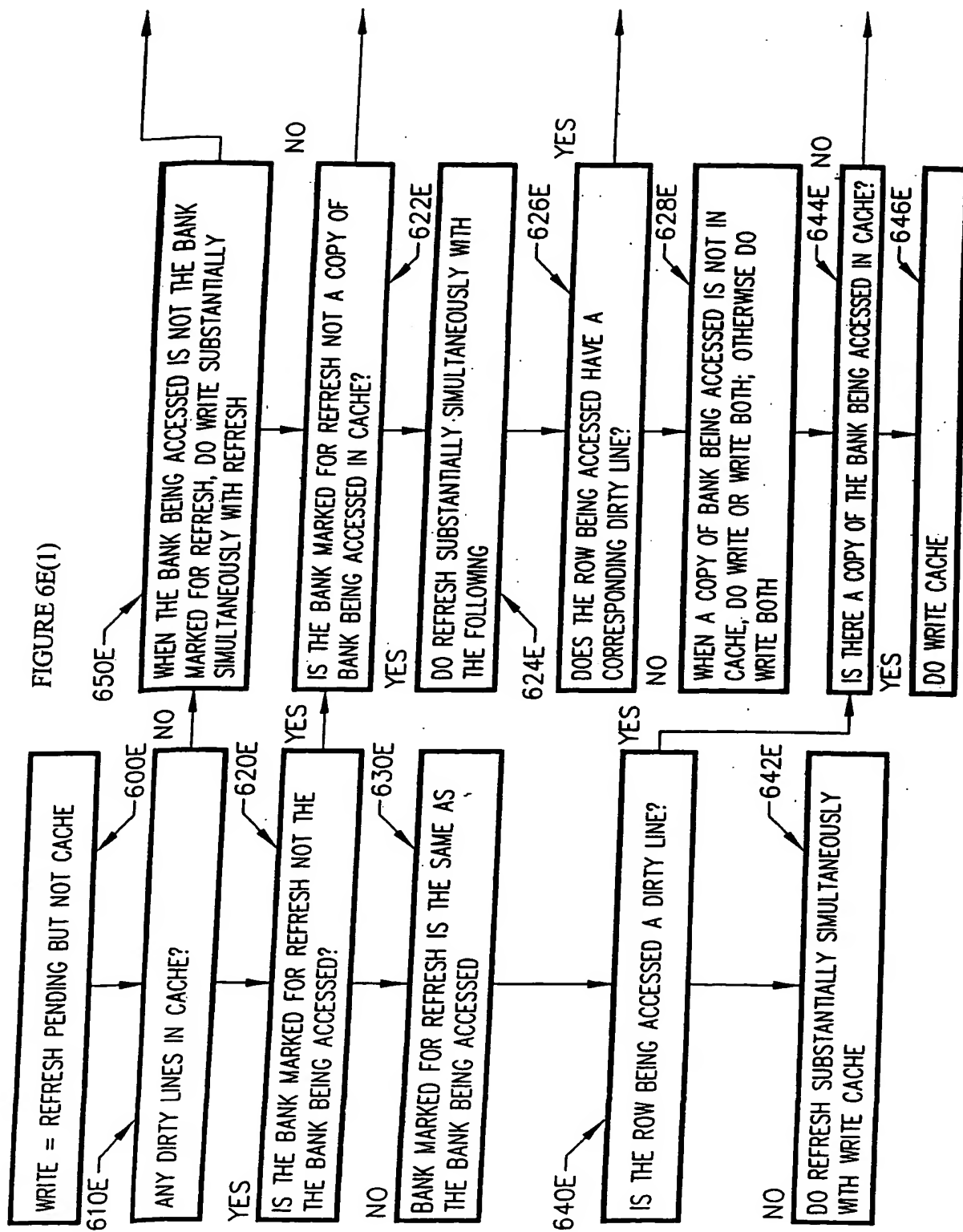


FIGURE 6E(2)

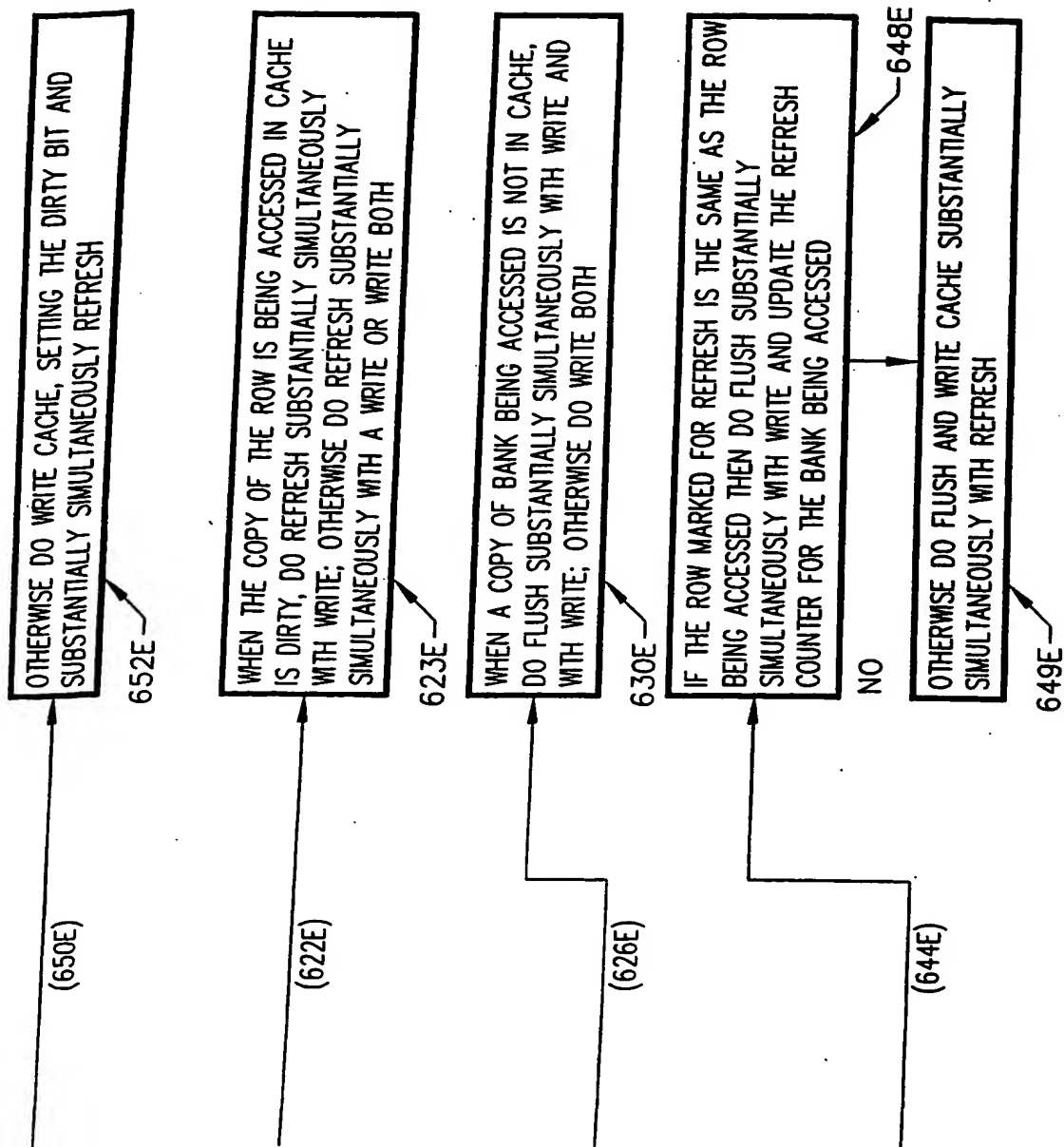


FIGURE 6F

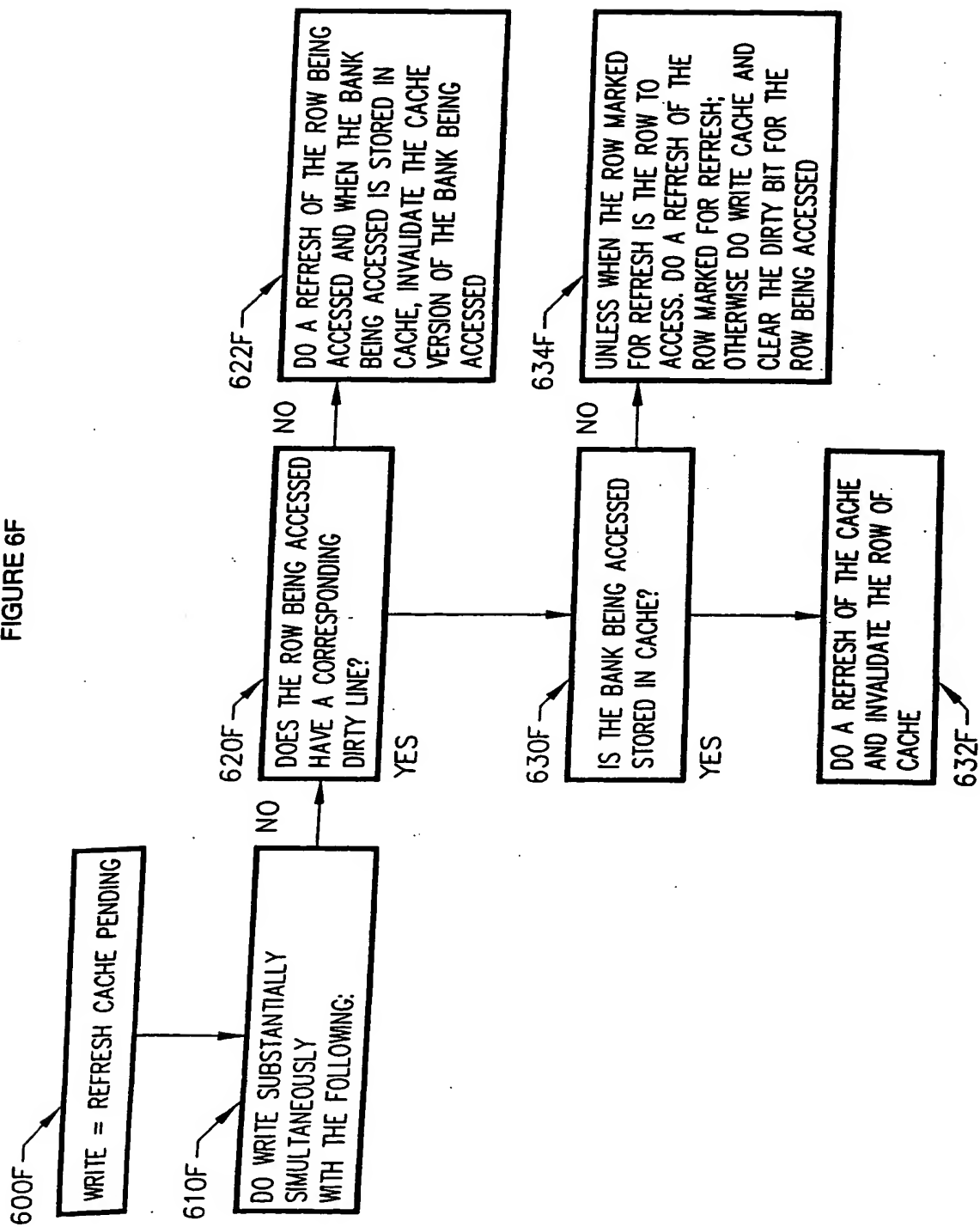


FIGURE 7A

## INPUTS

Refresh Counter: rb, rr /\* refresh bank, refresh row \*/  
 Current Access Address: wb, wr /\* access (write) bank and row \*/  
 Number of rows in a bank: row

## BUFFERS

Valid[row] /\* valid bits for each of the rows in the cache bank\*/  
 Dirty[row] /\* dirty bits for cache lines\*/  
 Cache[row] /\* cache bank \*/  
 Cachetag[row] /\* Tag for cache lines \*/

## FUNCTIONS

Refresh() {mem[rb, rr] = mem[rb, rr]; }  
 Flush() {mem[cachetag[wr], wr] = cache[wr] \$\$ dirty[wr] = 0;}  
 /\* WB dirty line \*/  
 ReadUpdateC() {data\_out = mem[wb, wr] \$\$ cache[wr] = mem[wb, wr]  
 \$\$ cachetag[wr] = wb; }  
 Read() {data\_out = mem[wb, wr]; }  
 Write() {mem[wb, wr] = data\_in; }  
 ReadCache() {data\_out = cache[wr] \$\$ cache[wr] = cache[wr];}  
 WriteCache() {cache[wr] = data\_in \$\$ cachetag[wr] = wb \$\$ dirty[wr] = 1;}  
 WriteBoth() {Write() \$\$ Writecache()\$\$ dirty[wr] = 0; }  
 /\* write to both the cache and memory \*/

(\$\$ - means operations are done in parallel)

## PSEUDO PROGRAM DESCRIPTION

/\* Initialize all cachetags to be NULL \*/  
 Cachetag[\*] = NULL;

/\*\*\* READ \*\*\*/

/\*----- No refresh pending ----- \*/  
 if (no refresh pending) {  
 if (no dirty line) {ReadUpdateC(); }  
 else /\* there is a dirty line \*/ {  
 if (!dirty[wr]) { ReadUpdateC();}  
 else if (wb == cachetag[wr]) { ReadCache() \$\$ Flush() \$\$ dirty[wr] = 0; }  
 else { Read() \$\$ Flush(); }  
 }  
 }



FIGURE 7B

```

/* -----Refresh pending -----*/
else {
    if (rb != cache) {
        if (no dirty line) {
            if (rb != wb) { Read() $$ Refresh(); }
            else { /* rb == wb */
                if (wb == tag[wr])
                    { ReadCache() $$ Refresh(); }
                else {
                    if (wr == rr) ReadUpdateC();
                    else Block_refresh $$ ReadUpdateC();
                }
            }
        }
        else /* there is a dirty line */
            if (rb != wb) {
                Refresh() $$
                if (tag[wr] != wb) {
                    if (dirty[wr]) { Flush() $$ Read(); }
                    else ReadUpdateC();
                }
                else {
                    if (dirty[wr]) { ReadCache() $$ Flush(); }
                    else Read();
                }
            }
        } /* end if rb != wb */
    else { /* refresh bank is the same as the accessed bank */
        if (wr == rr) {
            if (tag[wr] != wb) {
                if (dirty[wr]) { Flush() $$ Read()
                    $$ UpdateRefreshCounter(); }
                else { ReadUpdateC() $$ Update Refresh
                    Cntr(); }
            }
            else {
                if (dirty[wr]) { ReadCache() $$ Flush()
                    $$ UpdaterefreshCntr; }
                else { ReadUpdateC() $$
                    UpdateRefreshCounter(); }
            }
        }
        else {
            if (tag[wr] == wb) { Refresh() $$ ReadCache(); }
            /* dirty status for the line unchanged */
            else {
                BlockRefresh;
                if (dirty[wr]) { Flush() $$ Read(); }
                else ReadUpdateC();
            }
        }
    }
} /* end else there is a dirty line */
} /* end if rb != cache */

```

**FIGURE 7C**

```
else /* the cache bank is being refreshed */ {  
    if (dirty[wr]) {  
        if (tag[wr] == wb) {  
            BlockRefresh;  
            ReadCache() $$ Flush();  
        }  
        else {  
            BlockRefresh;  
            Read() $$ Flush();  
        }  
    }  
    else {  
        Refresh()  
        Read()  
    }  
}  
} /* end refresh pending */
```

FIGURE 7D

```
/** WRITE **/  
  
/* ----- No pending refresh ----- */  
if (no refresh pending) {  
    /* there is no dirty line */  
    if (no dirty line) {Write();}  
    else { /* there is a dirty line */  
        if (!dirty[wr]) { Write();}  
        else {  
            if (Cachetag[wr] == wb) { WriteBoth() $$ dirty[wr] = 0; }  
            else {Write() $$ Flush();}  
        }  
    } /* end dirty line */  
}
```

FIGURE 7E

```

/* ----- Pending refresh ----- */
else {
    if (rb != cache) {
        if (no dirty line) {
            if (wb != rb) { Write() $$ Refresh() }
            else { WriteCache() $$ dirty[wr] = 1 $$ Refresh() ;}
        }
        else { /* there is a dirty line */
            if (rb != wb) { /* refresh bank is different from the access bank */
                if (rb != cachetag[wr]) {
                    Refresh() $$
                    If (dirty[wr]) {
                        If (cachetag[wr] != wb) { Flush() $$ Write();}
                        Else { Writeboth();}
                    }
                }
                else {
                    If (cachetag[wr] != wb) { Write();} /* or writeboth() */
                    Else { Writeboth();}
                }
            }
            else {
                if (dirty[wr]) { Refresh() $$ Write();} /* cache line remains dirty */
                else { Refresh() $$ Write() ;} /* Or writeboth() */
            }
        }
    }
    else { /* refresh bank is the same as the access bank */
        if (dirty[wr]) {
            If (cachetag[wr] == wb) { Writecache();}
            else {
                if (wr == rr) { /* lucky day */
                    Flush() $$ write() $$ Updaterefreshcounter;
                }
                else {
                    Flush() and write to cache with new data ;
                    $$ Refresh();
                }
            }
        }
        else { /* the line is not dirty */
            /* create another dirty line */
            Refresh $$
            WriteCache();
        }
    }
}
}

```

FIGURE 7F

```
else { /* cache bank is being refreshed */
    Write() $$
    if (dirty[wr]) {
        if (cachetag[wr] == wb) {
            Refresh() $$
            Cachetag[wr] = NULL;
        }
        else {
            Refresh();
            /* unless wr = rr then we writeback and clear dirty bit */
        }
    }
    else { /* not dirty */
        Refresh();
        if (cachetag[wr] == wb) { Cachetag[wr] = NULL;}
    }
}
}
```

FIGURE 8

